UNIT II

Bitcoin and cryptocurrency

Bitcoin is a cryptocurrency developed by an unknown person or a group of unknown persons using the name Satoshi Nakamoto. It was first used in 2009 after being released as an open-source software.

What is Cryptocurrency?

Before going into detail about Bitcoin, it is imperative to understand what is a cryptocurrency. A cryptocurrency is a digital asset that functions as a medium of exchange wherein individual coin ownership records are stored in a ledger existing in a form of a computerized database using strong cryptography to secure transaction records.

Cryptocurrencies typically use decentralized control as opposed to centralized digital currency and central banking systems.

Definition of Cryptocurrency

In simplistic terms, Cryptocurrency is a digitized asset spread through multiple computers in a shared network. The decentralized nature of this network shields them from any control from government regulatory bodies.

The term “cryptocurrency in itself is derived from the encryption techniques used to secure the network.

As per computer experts, any system that falls under the category of cryptocurrency must meet the following requirements.:

1. Absence of any centralized authority and is maintained through distributed networks
2. The system maintains records of cryptocurrency units and who owns them
3. The system decides whether new units can be created and in case it does, decided the origin and the ownership terms
4. Ownership of cryptocurrency units can be proved exclusively cryptographically.
5. The system allows transactions to be performed in which ownership of the cryptographic units is changed.

OR

Cryptocurrency is a digital payment system that doesn't rely on banks to verify transactions. It is a peer-to-peer system that can enable anyone anywhere to send and receive payments. Instead of being physical money carried around and exchanged in the real world, cryptocurrency payments exist purely as digital entries to an online database describing specific transactions. When you transfer cryptocurrency funds, the transactions are recorded in a public ledger. Cryptocurrency is stored in digital wallets.

Cryptocurrency received its name because it uses encryption to verify transactions. This means advanced coding is involved in storing and transmitting cryptocurrency data between wallets and to public ledgers. The aim of encryption is to provide security and safety.
The first cryptocurrency was Bitcoin, which was founded in 2009 and remains the best known today. Much of the interest in cryptocurrencies is to trade for profit, with speculators at times driving prices skyward.

How does cryptocurrency work?

Cryptocurrencies run on a distributed public ledger called blockchain, a record of all transactions updated and held by currency holders.

Units of cryptocurrency are created through a process called mining, which involves using computer power to solve complicated mathematical problems that generate coins. Users can also buy the currencies from brokers, then store and spend them using cryptographic wallets.

If you own cryptocurrency, you don't own anything tangible. What you own is a key that allows you to move a record or a unit of measure from one person to another without a trusted third party.

Although Bitcoin has been around since 2009, cryptocurrencies and applications of blockchain technology are still emerging in financial terms, and more uses are expected in the future. Transactions including bonds, stocks, and other financial assets could eventually be traded using the technology.

Cryptocurrency examples

There are thousands of cryptocurrencies. Some of the best known include:

Bitcoin:

Founded in 2009, Bitcoin was the first cryptocurrency and is still the most commonly traded. The currency was developed by Satoshi Nakamoto – widely believed to be a pseudonym for an individual or group of people whose precise identity remains unknown.

Ethereum:

Developed in 2015, Ethereum is a blockchain platform with its own cryptocurrency, called Ether (ETH) or Ethereum. It is the most popular cryptocurrency after Bitcoin.

Litecoin:

This currency is most similar to bitcoin but has moved more quickly to develop new innovations, including faster payments and processes to allow more transactions.

Ripple:

Ripple is a distributed ledger system that was founded in 2012. Ripple can be used to track different kinds of transactions, not just cryptocurrency. The company behind it has worked with various banks and financial institutions.

Non-Bitcoin cryptocurrencies are collectively known as “altcoins” to distinguish them from the original.

How to buy cryptocurrency

You may be wondering how to buy cryptocurrency safely. There are typically three steps involved. These are:

Step 1: Choosing a platform

The first step is deciding which platform to use. Generally, you can choose between a traditional broker or dedicated cryptocurrency exchange:

- **Traditional brokers.** These are online brokers who offer ways to buy and sell cryptocurrency, as well as other financial assets like stocks, bonds, and ETFs. These platforms tend to offer lower trading costs but fewer crypto features.
- **Cryptocurrency exchanges.** There are many cryptocurrency exchanges to choose from, each offering different cryptocurrencies, wallet storage, interest-bearing account options, and more. Many exchanges charge asset-based fees.

When comparing different platforms, consider which cryptocurrencies are on offer, what fees they charge, their security features, storage and withdrawal options, and any educational resources.

Step 2: Funding your account

Once you have chosen your platform, the next step is to fund your account so you can begin trading. Most crypto exchanges allow users to purchase crypto using fiat (i.e., government-issued) currencies such as the US Dollar, the British Pound, or the Euro using their debit or credit cards – although this varies by platform.

Crypto purchases with credit cards are considered risky, and some exchanges don't support them. Some credit card companies don't allow crypto transactions either. This is because cryptocurrencies are highly volatile, and it is not advisable to risk going into debt — or potentially paying high credit card transaction fees — for certain assets.

Some platforms will also accept ACH transfers and wire transfers. The accepted payment methods and time taken for deposits or withdrawals differ per platform. Equally, the time taken for deposits to clear varies by payment method.

An important factor to consider is fees. These include potential deposit and withdrawal transaction fees plus trading fees. Fees will vary by payment method and platform, which is something to research at the outset.

Step 3: Placing an order

You can place an order via your broker's or exchange's web or mobile platform. If you are planning to buy cryptocurrencies, you can do so by selecting "buy," choosing the order type, entering the amount of cryptocurrencies you want to purchase, and confirming the order. The same process applies to "sell" orders.

There are also other ways to invest in crypto. These include payment services like PayPal, Cash App, and Venmo, which allow users to buy, sell, or hold cryptocurrencies. In addition, there are the following investment vehicles:

- **Bitcoin trusts:** You can buy shares of Bitcoin trusts with a regular brokerage account. These vehicles give retail investors exposure to crypto through the stock market.
- **Bitcoin mutual funds:** There are Bitcoin ETFs and Bitcoin mutual funds to choose from.
- **Blockchain stocks or ETFs:** You can also indirectly invest in crypto through blockchain companies that specialize in the technology behind crypto and crypto transactions. Alternatively, you can buy stocks or ETFs of companies that use blockchain technology.

The best option for you will depend on your investment goals and risk appetite.

How to store cryptocurrency

Once you have purchased cryptocurrency, you need to store it safely to protect it from hacks or theft. Usually, cryptocurrency is stored in crypto wallets, which are physical devices or online software used to store the private keys to your cryptocurrencies securely. Some exchanges provide wallet services, making it easy for you to store directly through the platform. However, not all exchanges or brokers automatically provide wallet services for you.

There are different wallet providers to choose from. The terms “hot wallet” and “cold wallet” are used:

- **Hot wallet storage:** "hot wallets" refer to crypto storage that uses online software to protect the private keys to your assets.
- **Cold wallet storage:** Unlike hot wallets, cold wallets (also known as hardware wallets) rely on offline electronic devices to securely store your private keys.

Typically, cold wallets tend to charge fees, while hot wallets don't.

Types of Cryptocurrency

The first type of crypto currency was Bitcoin, which to this day remains the most-used, valuable and popular. Along with Bitcoin, other alternative cryptocurrencies with varying degrees of functions and specifications have been created. Some are iterations of bitcoin while others have been created from the ground up

Bitcoin was launched in 2009 by an individual or group known by the pseudonym “Satoshi Nakamoto. As of March 2021, there were over 18.6 million bitcoins in circulation with a total market cap of around $927 billion.

The competing cryptocurrencies that were created as a result of Bitcoin’s success are known as altcoins. Some of the well-known altcoins are as follows:

1. Litecoin
2. Peercoin
3. Namecoin
4. Ethereum
5. Cardana

Today, the aggregate value of all the cryptocurrencies in existence is around $1.5 trillion—Bitcoin currently represents more than 60% of the total value.3.

Advantages and disadvantages of Cryptocurrency

Cryptocurrency has the following advantages

- Funds transfer between two parties will be easy without the need of third party like credit/debit cards or banks
- It is a cheaper alternative compared to other online transactions
- Payments are safe and secured and offer an unprecedented level of anonymity
- Modern cryptocurrency systems come with a user "wallet" or account address which is accessible only by a public key and pirate key. The private key is only know to the owner of the wallet
- Funds transfer are completed with minimal processing fees.

Cryptocurrencies have the following disadvantages.

- The almost hidden nature of cryptocurrency transactions makes them easy to be the focus of illegal activities such as money laundering, tax-evasion and possibly even terror-financing
- Payments are not irreversible
- Cryptocurrencies are not accepted everywhere and have limited value elsewhere
- There is concern that cryptocurrencies like Bitcoin are not rooted in any material goods. Some research, however, has identified that the cost of producing a Bitcoin, which requires an increasingly large amount of energy, is directly related to its market price.

What can you buy with cryptocurrency?

When it was first launched, Bitcoin was intended to be a medium for daily transactions, making it possible to buy everything from a cup of coffee to a computer or even big-ticket items like real estate. That hasn't quite materialized and, while the number of institutions accepting cryptocurrencies is growing, large transactions involving it are rare. Even so, it is possible to buy a wide variety of products from e-commerce websites using crypto. Here are some examples:

Technology and e-commerce sites:

Several companies that sell tech products accept crypto on their websites, such as newegg.com, AT&T, and Microsoft. Overstock, an e-commerce platform, was among the first sites to accept Bitcoin. Shopify, Rakuten, and Home Depot also accept it.

Luxury goods:

Some luxury retailers accept crypto as a form of payment. For example, online luxury retailer Bitdials offers Rolex, Patek Philippe, and other high-end watches in return for Bitcoin.

Cars:

Some car dealers – from mass-market brands to high-end luxury dealers – already accept cryptocurrency as payment.

Insurance:
In April 2021, Swiss insurer AXA announced that it had begun accepting Bitcoin as a mode of payment for all its lines of insurance except life insurance (due to regulatory issues). Premier Shield Insurance, which sells home and auto insurance policies in the US, also accepts Bitcoin for premium payments.

If you want to spend cryptocurrency at a retailer that doesn't accept it directly, you can use a cryptocurrency debit card, such as BitPay in the US.

Cryptocurrency fraud and cryptocurrency scams

Unfortunately, cryptocurrency crime is on the rise. Cryptocurrency scams include:

Fake websites: Bogus sites which feature fake testimonials and crypto jargon promising massive, guaranteed returns, provided you keep investing.
Virtual Ponzi schemes: Cryptocurrency criminals promote non-existent opportunities to invest in digital currencies and create the illusion of huge returns by paying off old investors with new investors' money. One scam operation, BitClub Network, raised more than $700 million before its perpetrators were indicted in December 2019.
"Celebrity" endorsements: Scammers pose online as billionaires or well-known names who promise to multiply your investment in a virtual currency but instead steal what you send. They may also use messaging apps or chat rooms to start rumours that a famous businessperson is backing a specific cryptocurrency. Once they have encouraged investors to buy and driven up the price, the scammers sell their stake, and the currency reduces in value.
Romance scams: The FBI warns of a trend in online dating scams, where tricksters persuade people they meet on dating apps or social media to invest or trade in virtual currencies. The FBI's Internet Crime Complaint Centre fielded more than 1,800 reports of crypto-focused romance scams in the first seven months of 2021, with losses reaching $133 million.

Otherwise, fraudsters may pose as legitimate virtual currency traders or set up bogus exchanges to trick people into giving them money. Another crypto scam involves fraudulent sales pitches for individual retirement accounts in cryptocurrencies. Then there is straightforward cryptocurrency

hacking, where criminals break into the digital wallets where people store their virtual currency to steal it.

Is cryptocurrency safe?

Cryptocurrencies are usually built using blockchain technology. Blockchain describes the way transactions are recorded into "blocks" and time stamped. It's a fairly complex, technical process, but the result is a digital ledger of cryptocurrency transactions that's hard for hackers to tamper with.

In addition, transactions require a two-factor authentication process. For instance, you might be asked to enter a username and password to start a transaction. Then, you might have to enter an authentication code sent via text to your personal cell phone.

While securities are in place, that does not mean cryptocurrencies are un-hackable. Several high-dollar hacks have cost cryptocurrency start-ups heavily. Hackers hit Coincheck to the tune of $534 million and BitGrail for $195 million, making them two of the biggest cryptocurrency hacks of 2018.

Unlike government-backed money, the value of virtual currencies is driven entirely by supply and demand. This can create wild swings that produce significant gains for investors or big losses. And cryptocurrency investments are subject to far less regulatory protection than traditional financial products like stocks, bonds, and mutual funds.

What is Bitcoin?

- Bitcoin is the maiden implementation of a concept known as “cryptocurrency”. This concept was first described by Wei Dai in the year 1998 on the cypherpunks mailing list wherein he proposed the idea of a new form of money that uses cryptography to control its creation and transactions, rather than with a central authority.
- Bitcoin (BTC) is a cryptocurrency, a virtual currency designed to act as money and a form of payment outside the control of any one person, group, or entity, thus removing the need for third-party involvement in financial transactions. It is rewarded to blockchain miners for the work done to verify transactions and can be purchased on several exchanges.
- Satoshi Nakamoto, about whom nothing much is known, was the first person to give Bitcoin specifications and also provide proof of the concept, which he did in 2009. He provided these in a cryptography mailing list. In 2010, Satoshi quit the project. Since then, the community has multiplied with currently many developers working on Bitcoin.
- It has since become the most well-known cryptocurrency in the world. Its popularity has inspired the development of many other cryptocurrencies. These competitors either attempt to replace it as a payment system or are used as utility or security tokens in other blockchains and emerging financial technologies.

KEY Points

- Launched in 2009, Bitcoin is the world's largest cryptocurrency by market capitalization.
- Unlike fiat currency, Bitcoin is created, distributed, traded, and stored using a decentralized ledger system known as a blockchain.
- Bitcoin and its ledger are secured by proof-of-work (PoW) consensus, which is also the "mining" process that introduces new bitcoins into the system.
- Bitcoin can be purchased via various cryptocurrency exchanges.
- Bitcoin's history as a store of value has been turbulent; it has gone through several boom and bust cycles over its relatively short lifespan.
- As the first decentralized virtual currency to meet widespread popularity and success, Bitcoin has inspired a host of other cryptocurrencies in its wake.

Bitcoin's Blockchain Technology

Cryptocurrencies are part of a blockchain and the network required to power it. A blockchain is a distributed ledger, a shared database that stores data. Data within the blockchain is secured by encryption methods.

When a transaction takes place on the blockchain, information from the previous block is copied to a new block with the new data, encrypted, and the transaction is verified by validators—called miners—in the network. When a transaction is verified, a new block is opened, and a Bitcoin is created and given as a reward to the miner(s) who verified the data within the block—they are then free to use it, hold it, or sell it.

Bitcoin uses the SHA-256 hashing algorithm to encrypt the data stored in the blocks on the blockchain. Simply put, transaction data stored in a block is encrypted into a 256-bit hexadecimal number. That number contains all of the transaction data and information linked to the blocks before that block.

Data linked between blocks is what led to the ledger being called a blockchain.

Transactions are placed into a queue to be validated by miners within the network. Miners in the Bitcoin blockchain network all attempt to verify the same transaction simultaneously. The mining software and hardware work to solve the nonce, a four-byte number included in the block header that miners are attempting to solve.

The block header is hashed, or randomly regenerated by a miner repeatedly until it meets a target number specified by the blockchain. The block header is "solved," and a new block is created for more transactions to be encrypted and verified.

Who controls bitcoin?

- The Bitcoin network is owned by nobody quite like how the email technology is not owned by anyone.

- All Bitcoin users all over the globe control Bitcoin. Developers can improve on the software but they cannot enforce a change in its protocol. This is because all the users have the freedom to opt for the software and version they wish to use.
- For staying compatible with one another, all users have to use software that complies with the same rules.
- Because Bitcoin can work accurately only by a complete consensus among all its users, there is a strong motivation among its users to protect this consensus.

Payments by Bitcoin

- Payments by Bitcoin are simpler to make when compared to a credit or debit card transaction. They can also be received with no merchant account.
- Payments can be done from a wallet application (on a smartphone or a computer) by entering the address of the recipient, the payment sum, and press the send button.
- To make it easier to enter a receiver's address, many wallets can get the address by scanning a QR code or by touching two phones together with NFC technology.
- Bitcoin can be used like any other money form either online or in a brick-and-mortar store.

Bitcoin in India

In March 2021, it was announced by the Government of India that it would be introducing legislation that would penalise an individual found to be in the possession of bitcoin.

The legislation, one of the world's strictest policies against cryptocurrencies, would criminalise possession, issuance, mining, trading and transferring crypto-assets, said the official, who has direct knowledge of the plan.

The measure is in line with a January government agenda that called for banning private virtual currencies such as bitcoin while building a framework for an official digital currency.

The Bitcoin Network

Peer-to-Peer Network Architecture

Bitcoin is structured as a peer-to-peer network architecture on top of the Internet. The term peer-to-peer, or P2P, means that the computers that participate in the network are peers to each other, that they are all equal, that there are no "special" nodes, and that all nodes share the burden of providing network services. The network nodes interconnect in a mesh network with a "flat" topology. There is no server, no centralized service, and no hierarchy within the network. Nodes in a peer-to-peer network both provide and consume services at the same time with reciprocity acting as the incentive for participation. Peer-to-peer networks are inherently resilient, decentralized, and open. The preeminent example of a P2P network architecture was the early

Internet itself, where nodes on the IP network were equal. Today's Internet architecture is more hierarchical, but the Internet Protocol still retains its flat-topology essence. Beyond bitcoin, the largest and most successful application of P2P technologies is file sharing with Napster as the pioneer and BitTorrent as the most recent evolution of the architecture.

Bitcoin's P2P network architecture is much more than a topology choice. Bitcoin is a peer-to-peer digital cash system by design, and the network architecture is both a reflection and a foundation of that core characteristic. Decentralization of control is a core design principle and that can only be achieved and maintained by a flat, decentralized P2P consensus network.

The term "bitcoin network" refers to the collection of nodes running the bitcoin P2P protocol. In addition to the bitcoin P2P protocol, there are other protocols such as Stratum, which are used for mining and lightweight or mobile wallets. These additional protocols are provided by gateway routing servers that access the bitcoin network using the bitcoin P2P protocol, and then extend that network to nodes running other protocols. For example, Stratum servers connect Stratum mining nodes via the Stratum protocol to the main bitcoin network and bridge the Stratum protocol to the bitcoin P2P protocol. We use the term "extended bitcoin network" to refer to the overall network that includes the bitcoin P2P protocol, pool-mining protocols, the Stratum protocol, and any other related protocols connecting the components of the bitcoin system.

Nodes Types and Roles

Although nodes in the bitcoin P2P network are equal, they may take on different roles depending on the functionality they are supporting. A bitcoin node is a collection of functions: routing, the blockchain database, mining, and wallet services. A full node with all four of these functions is shown in Figure.

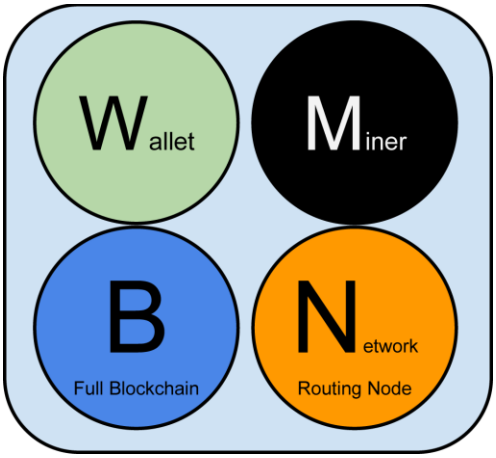

Figure 1 :- A bitcoin network node with all four functions: wallet, miner, full blockchain database, and network routing

All nodes include the routing function to participate in the network and might include other functionality. All nodes validate and propagate transactions and blocks, and discover and maintain connections to peers. In the full-node example in Figure 1, the routing function is indicated by an orange circle named "Network Routing Node."

Some nodes, called full nodes, also maintain a complete and up-to-date copy of the blockchain. Full nodes can autonomously and authoritatively verify any transaction without external reference. Some nodes maintain only a subset of the blockchain and verify transactions using a method called *simplified payment verification*, or SPV. These nodes are known as SPV or lightweight nodes. In the full-node example in the figure, the full-node blockchain database function is indicated by a blue circle named "Full Blockchain." In Figure 3, SPV nodes are drawn without the blue circle, showing that they do not have a full copy of the blockchain.

Mining nodes compete to create new blocks by running specialized hardware to solve the proof-of-work algorithm. Some mining nodes are also full nodes, maintaining a full copy of the blockchain, while others are lightweight nodes participating in pool mining and depending on a pool server to maintain a full node. The mining function is shown in the full node as a black circle named "Miner."

User wallets might be part of a full node, as is usually the case with desktop bitcoin clients. Increasingly, many user wallets, especially those running on resource-constrained devices such as smartphones, are SPV nodes. The wallet function is shown in Figure 1 as a green circle named "Wallet".

In addition to the main node types on the bitcoin P2P protocol, there are servers and nodes running other protocols, such as specialized mining pool protocols and lightweight client-access protocols.

Figure 2 shows the most common node types on the extended bitcoin network.

The Extended Bitcoin Network

The main bitcoin network, running the bitcoin P2P protocol, consists of between 7,000 and 10,000 listening nodes running various versions of the bitcoin reference client (Bitcoin Core) and a few hundred nodes running various other implementations of the bitcoin P2P protocol, such as BitcoinJ, Libbitcoin, and btcd. A small percentage of the nodes on the bitcoin P2P network are

also mining nodes, competing in the mining process, validating transactions, and creating new blocks. Various large companies interface with the bitcoin network by running full-node clients based on the Bitcoin Core client, with full copies of the blockchain and a network node, but without mining or wallet functions. These nodes act as network edge routers, allowing various other services (exchanges, wallets, block explorers, merchant payment processing) to be built on top.

The extended bitcoin network includes the network running the bitcoin P2P protocol, described earlier, as well as nodes running specialized protocols. Attached to the main bitcoin P2P network are a number of pool servers and protocol gateways that connect nodes running other protocols. These other protocol nodes are mostly pool mining nodes and lightweight wallet clients, which do not carry a full copy of the blockchain.

Figure 3 shows the extended bitcoin network with the various types of nodes, gateway servers, edge routers, and wallet clients and the various protocols they use to connect to each other.

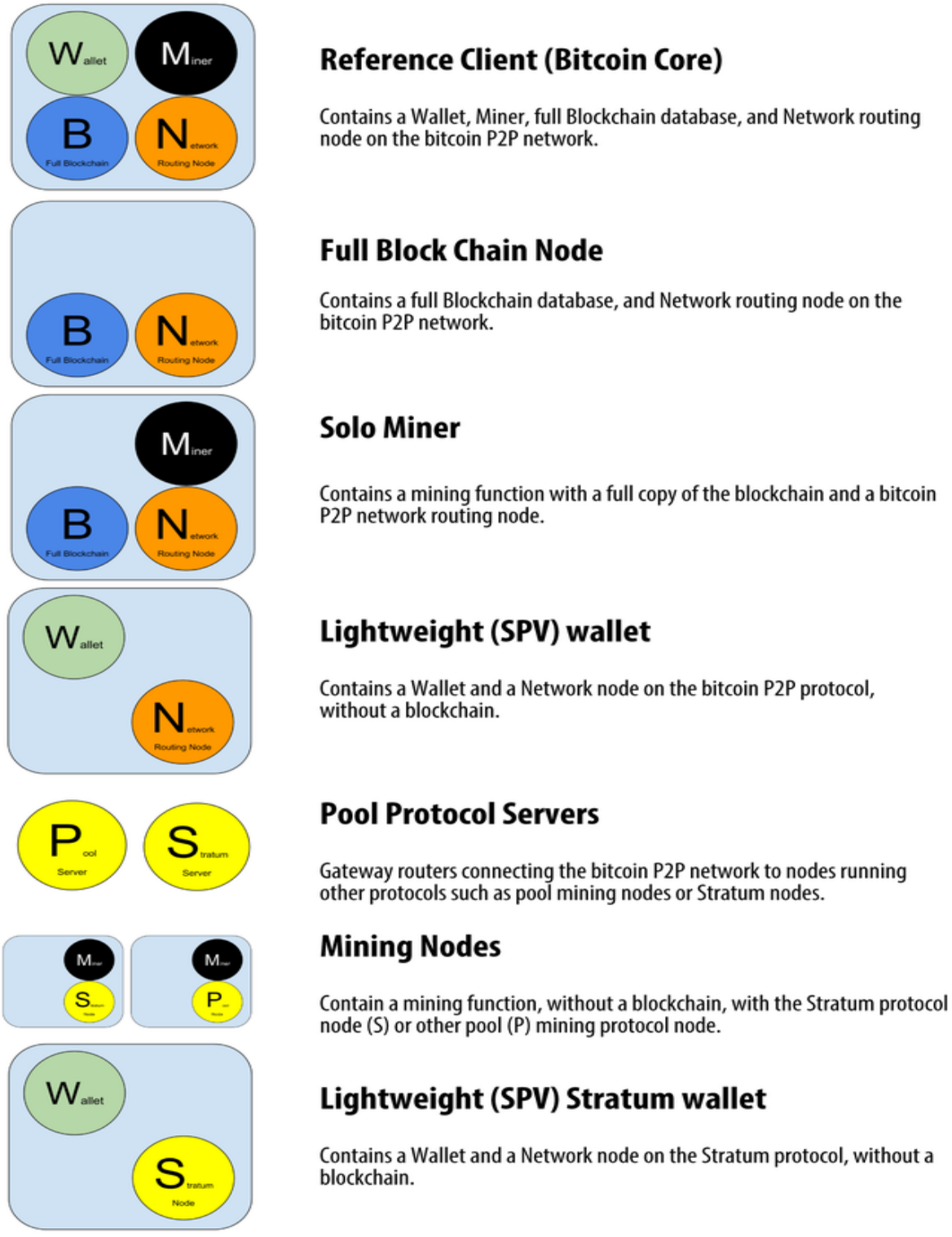

Figure 2. Different types of nodes on the extended bitcoin network

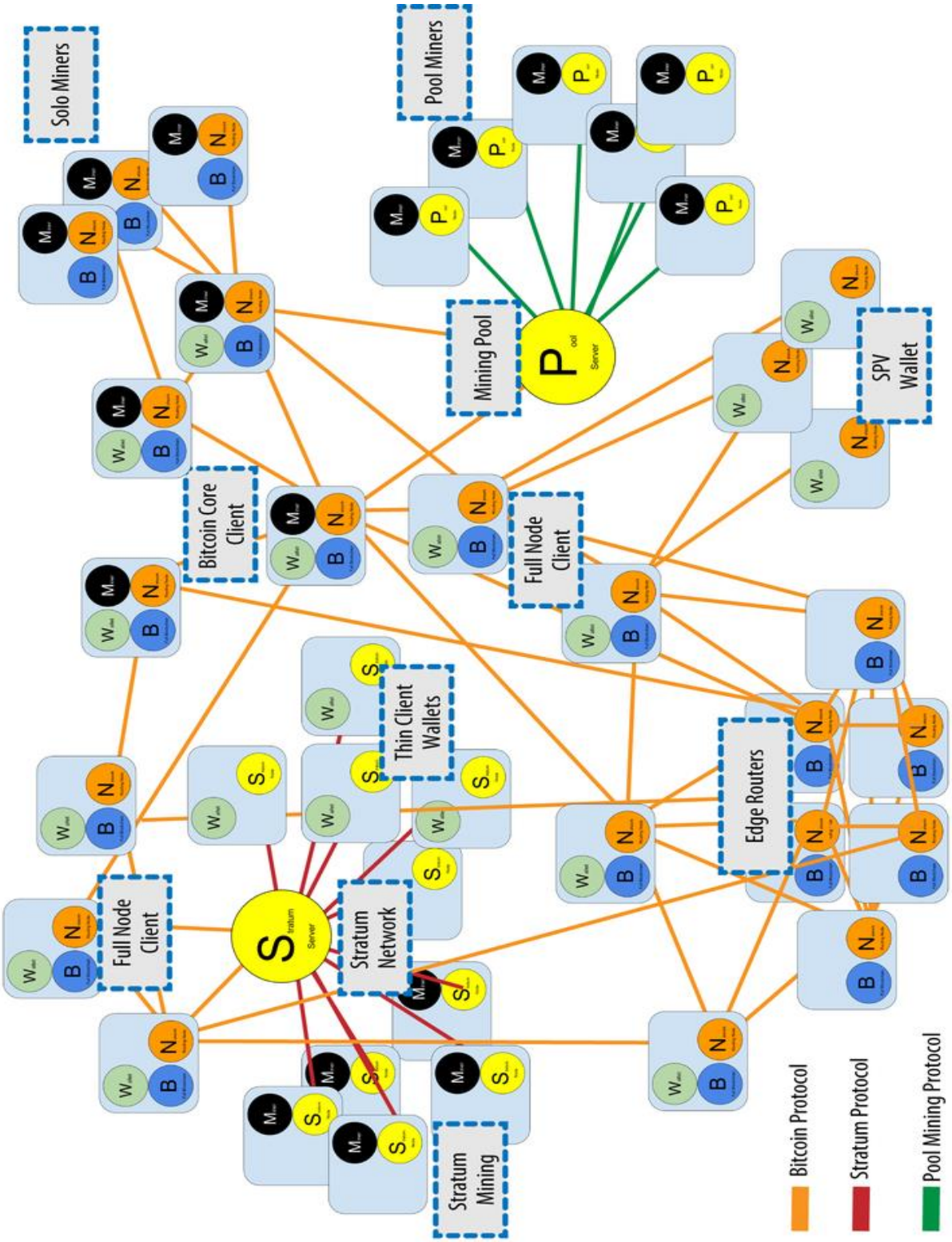

Figure 3. The extended bitcoin network showing various node types, gateways, and protocols

Network Discovery

When a new node boots up, it must discover other bitcoin nodes on the network in order to participate. To start this process, a new node must discover at least one existing node on the network and connect to it. The geographic location of other nodes is irrelevant; the bitcoin network topology is not geographically defined. Therefore, any existing bitcoin nodes can be selected at random.

To connect to a known peer, nodes establish a TCP connection, usually to port 8333 (the port generally known as the one used by bitcoin), or an alternative port if one is provided. Upon establishing a connection, the node will start a "handshake" (see Figure 4) by transmitting a version message, which contains basic identifying information, including:

PROTOCOL_VERSION
: A constant that defines the bitcoin P2P protocol version the client "speaks" (e.g., 70002)

nLocalServices
: A list of local services supported by the node, currently just NODE_NETWORK

nTime
: The current time

addrYou
: The IP address of the remote node as seen from this node

addrMe
: The IP address of the local node, as discovered by the local node

subver
: A sub-version showing the type of software running on this node (e.g., "/Satoshi:0.9.2.1/")+

BestHeight
: The block height of this node's blockchain

The peer node responds with verack to acknowledge and establish a connection, and optionally sends its own version message if it wishes to reciprocate the connection and connect back as a peer.

How does a new node find peers? The first method is to query DNS using a number of "DNS seeds," which are DNS servers that provide a list of IP addresses of bitcoin nodes. Some of those DNS seeds provide a static list of IP addresses of stable bitcoin listening nodes. Some of the DNS seeds are custom implementations of BIND (Berkeley Internet Name Daemon) that return a

random subset from a list of bitcoin node addresses collected by a crawler or a long-running bitcoin node. The Bitcoin Core client contains the names of five different DNS seeds. The diversity of ownership and diversity of implementation of the different DNS seeds offers a high level or reliability for the initial bootstrapping process. In the Bitcoin Core client, the option to use the DNS seeds is controlled by the option switch -dnsseed (set to 1 by default, to use the DNS seed).

Alternatively, a bootstrapping node that knows nothing of the network must be given the IP address of at least one bitcoin node, after which it can establish connections through further introductions. The command-line argument -seednode can be used to connect to one node just for introductions, using it as a seed. After the initial seed node is used to form introductions, the client will disconnect from it and use the newly discovered peers.

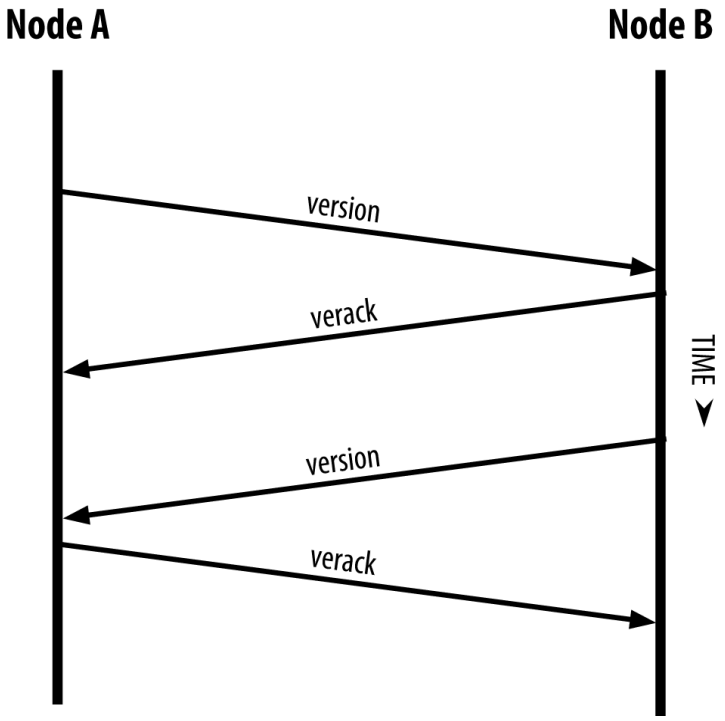

Figure 4. The initial handshake between peers

Once one or more connections are established, the new node will send an addr message containing its own IP address to its neighbors. The neighbors will, in turn, forward the addr message to their neighbors, ensuring that the newly connected node becomes well known and better connected. Additionally, the newly connected node can send getaddr to the neighbors, asking them to return a list of IP addresses of other peers. That way, a node can find peers to connect to and advertise its existence on the network for other nodes to find it. Figure 6-5 shows the address discovery protocol.

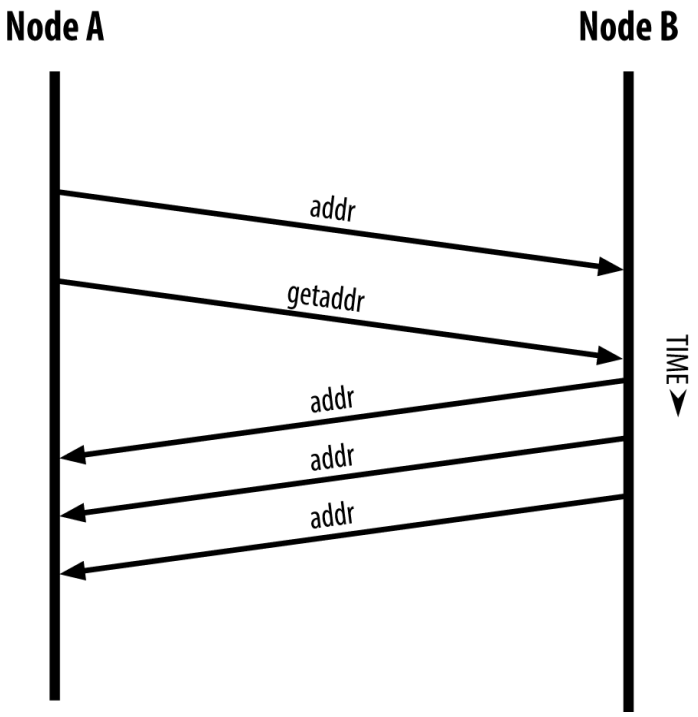

Figure 5. Address propagation and discovery

A node must connect to a few different peers in order to establish diverse paths into the bitcoin network. Paths are not reliable—nodes come and go—and so the node must continue to discover new nodes as it loses old connections as well as assist other nodes when they bootstrap. Only one connection is needed to bootstrap, because the first node can offer introductions to its peer nodes and those peers can offer further introductions. It's also unnecessary and wasteful of network resources to connect to more than a handful of nodes. After bootstrapping, a node will remember its most recent successful peer connections, so that if it is rebooted it can quickly reestablish connections with its former peer network. If none of the former peers respond to its connection request, the node can use the seed nodes to bootstrap again.

On a node running the Bitcoin Core client, you can list the peer connections with the command getpeerinfo:

```
$ bitcoin-cli getpeerinfo
[
  {
    "addr" : "85.213.199.39:8333",
    "services" : "00000001",
    "lastsend" : 1405634126,
    "lastrecv" : 1405634127,
```

```
        "bytessent" : 23487651,
        "bytesrecv" : 138679099,
        "conntime" : 1405021768,
        "pingtime" : 0.00000000,
        "version" : 70002,
        "subver" : "/Satoshi:0.9.2.1/",
        "inbound" : false,
        "startingheight" : 310131,
        "banscore" : 0,
        "syncnode" : true
    },
    {
        "addr" : "58.23.244.20:8333",
        "services" : "00000001",
        "lastsend" : 1405634127,
        "lastrecv" : 1405634124,
        "bytessent" : 4460918,
        "bytesrecv" : 8903575,
        "conntime" : 1405559628,
        "pingtime" : 0.00000000,
        "version" : 70001,
        "subver" : "/Satoshi:0.8.6/",
        "inbound" : false,
        "startingheight" : 311074,
        "banscore" : 0,
        "syncnode" : false
    }
]
```


To override the automatic management of peers and to specify a list of IP addresses, users can provide the option -connect=<IPAddress> and specify one or more IP addresses. If this option is used, the node will only connect to the selected IP addresses, instead of discovering and maintaining the peer connections automatically.

If there is no traffic on a connection, nodes will periodically send a message to maintain the connection. If a node has not communicated on a connection for more than 90 minutes, it is assumed to be disconnected and a new peer will be sought. Thus, the network dynamically adjusts

to transient nodes and network problems, and can organically grow and shrink as needed without any central control.

Full Nodes

Full nodes are nodes that maintain a full blockchain with all transactions. More accurately, they probably should be called "full blockchain nodes." In the early years of bitcoin, all nodes were full nodes and currently the Bitcoin Core client is a full blockchain node. In the past two years, however, new forms of bitcoin clients have been introduced that do not maintain a full blockchain but run as lightweight clients. We'll examine these in more detail in the next section.

Full blockchain nodes maintain a complete and up-to-date copy of the bitcoin blockchain with all the transactions, which they independently build and verify, starting with the very first block (genesis block) and building up to the latest known block in the network. A full blockchain node can independently and authoritatively verify any transaction without recourse or reliance on any other node or source of information. The full blockchain node relies on the network to receive updates about new blocks of transactions, which it then verifies and incorporates into its local copy of the blockchain.

Running a full blockchain node gives you the pure bitcoin experience: independent verification of all transactions without the need to rely on, or trust, any other systems. It's easy to tell if you're running a full node because it requires 20+ gigabytes of persistent storage (disk space) to store the full blockchain. If you need a lot of disk and it takes two to three days to sync to the network, you are running a full node. That is the price of complete independence and freedom from central authority.

There are a few alternative implementations of full blockchain bitcoin clients, built using different programming languages and software architectures. However, the most common implementation is the reference client Bitcoin Core, also known as the Satoshi client. More than 90% of the nodes on the bitcoin network run various versions of Bitcoin Core. It is identified as "Satoshi" in the sub-version string sent in the version message and shown by the command getpeerinfo as we saw earlier; for example, /Satoshi:0.8.6/.

Exchanging "Inventory"

The first thing a full node will do once it connects to peers is try to construct a complete blockchain. If it is a brand-new node and has no blockchain at all, it only knows one block, the genesis block,

which is statically embedded in the client software. Starting with block #0 (the genesis block), the new node will have to download hundreds of thousands of blocks to synchronize with the network and re-establish the full blockchain.

The process of syncing the blockchain starts with the version message, because that contains BestHeight, a node's current blockchain height (number of blocks). A node will see the version messages from its peers, know how many blocks they each have, and be able to compare to how many blocks it has in its own blockchain. Peered nodes will exchange a getblocks message that contains the hash (fingerprint) of the top block on their local blockchain. One of the peers will be able to identify the received hash as belonging to a block that is not at the top, but rather belongs to an older block, thus deducing that its own local blockchain is longer than its peer's.

The peer that has the longer blockchain has more blocks than the other node and can identify which blocks the other node needs in order to "catch up." It will identify the first 500 blocks to share and transmit their hashes using an inv (inventory) message. The node missing these blocks will then retrieve them, by issuing a series of getdata messages requesting the full block data and identifying the requested blocks using the hashes from the inv message.

Let's assume, for example, that a node only has the genesis block. It will then receive an inv message from its peers containing the hashes of the next 500 blocks in the chain. It will start requesting blocks from all of its connected peers, spreading the load and ensuring that it doesn't overwhelm any peer with requests. The node keeps track of how many blocks are "in transit" per peer connection, meaning blocks that it has requested but not received, checking that it does not exceed a limit (MAX_BLOCKS_IN_TRANSIT_PER_PEER). This way, if it needs a lot of blocks, it will only request new ones as previous requests are fulfilled, allowing the peers to control the pace of updates and not overwhelming the network. As each block is received, it is added to the blockchain. As the local blockchain is gradually built up, more blocks are requested and received, and the process continues until the node catches up to the rest of the network.

This process of comparing the local blockchain with the peers and retrieving any missing blocks happens any time a node goes offline for any period of time. Whether a node has been offline for a few minutes and is missing a few blocks, or a month and is missing a few thousand blocks, it starts by sending getblocks, gets an inv response, and starts downloading the missing blocks. Figure 6 shows the inventory and block propagation protocol.

Simplified Payment Verification (SPV) Nodes

Not all nodes have the ability to store the full blockchain. Many bitcoin clients are designed to run on space- and power-constrained devices, such as smartphones, tablets, or embedded systems. For such devices, a simplified payment verification (SPV) method is used to allow them to operate without storing the full blockchain. These types of clients are called SPV clients or lightweight clients. As bitcoin adoption surges, the SPV node is becoming the most common form of bitcoin node, especially for bitcoin wallets.

SPV nodes download only the block headers and do not download the transactions included in each block. The resulting chain of blocks, without transactions, is 1,000 times smaller than the full blockchain. SPV nodes cannot construct a full picture of all the UTXOs that are available for spending because they do not know about all the transactions on the network. SPV nodes verify transactions using a slightly different methodology that relies on peers to provide partial views of relevant parts of the blockchain on demand.

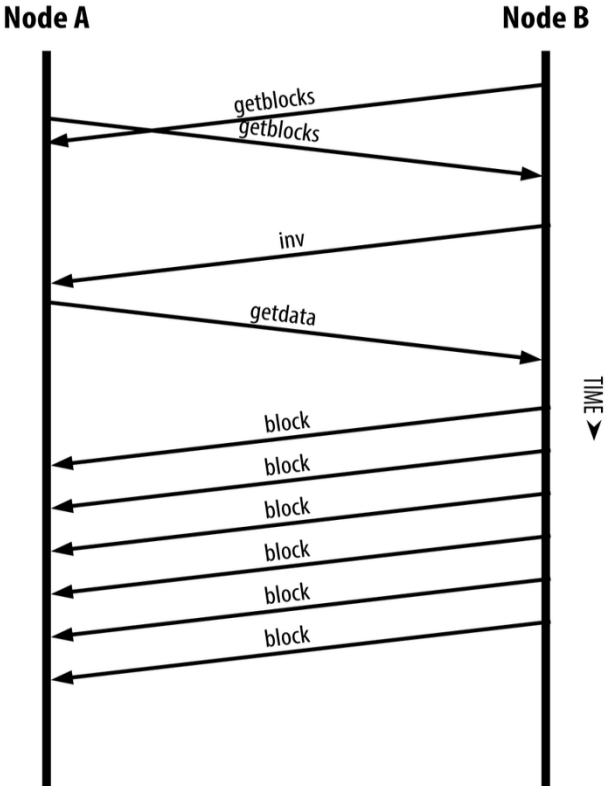

Figure 6. Node synchronizing the blockchain by retrieving blocks from a peer

As an analogy, a full node is like a tourist in a strange city, equipped with a detailed map of every street and every address. By comparison, an SPV node is like a tourist in a strange city asking random strangers for turn-by-turn directions while knowing only one main avenue. Although both

tourists can verify the existence of a street by visiting it, the tourist without a map doesn't know what lies down any of the side streets and doesn't know what other streets exist. Positioned in front of 23 Church Street, the tourist without a map cannot know if there are a dozen other "23 Church Street" addresses in the city and whether this is the right one. The mapless tourist's best chance is to ask enough people and hope some of them are not trying to mug him.

Simplified payment verification verifies transactions by reference to their *depth* in the blockchain instead of their *height*. Whereas a full blockchain node will construct a fully verified chain of thousands of blocks and transactions reaching down the blockchain (back in time) all the way to the genesis block, an SPV node will verify the chain of all blocks (but not all transactions) and link that chain to the transaction of interest.

For example, when examining a transaction in block 300,000, a full node links all 300,000 blocks down to the genesis block and builds a full database of UTXO, establishing the validity of the transaction by confirming that the UTXO remains unspent. An SPV node cannot validate whether the UTXO is unspent. Instead, the SPV node will establish a link between the transaction and the block that contains it, using a merkle path (see Merkle Trees). Then, the SPV node waits until it sees the six blocks 300,001 through 300,006 piled on top of the block containing the transaction and verifies it by establishing its depth under blocks 300,006 to 300,001. The fact that other nodes on the network accepted block 300,000 and then did the necessary work to produce six more blocks on top of it is proof, by proxy, that the transaction was not a double-spend.

An SPV node cannot be persuaded that a transaction exists in a block when the transaction does not in fact exist. The SPV node establishes the existence of a transaction in a block by requesting a merkle path proof and by validating the proof of work in the chain of blocks. However, a transaction's existence can be "hidden" from an SPV node. An SPV node can definitely prove that a transaction exists but cannot verify that a transaction, such as a double-spend of the same UTXO, doesn't exist because it doesn't have a record of all transactions. This vulnerability can be used in a denial-of-service attack or for a double-spending attack against SPV nodes. To defend against this, an SPV node needs to connect randomly to several nodes, to increase the probability that it is in contact with at least one honest node. This need to randomly connect means that SPV nodes also are vulnerable to network partitioning attacks or Sybil attacks, where they are connected to fake nodes or fake networks and do not have access to honest nodes or the real bitcoin network.

For most practical purposes, well-connected SPV nodes are secure enough, striking the right balance between resource needs, practicality, and security. For infallible security, however, nothing beats running a full blockchain node.

TIP

A full blockchain node verifies a transaction by checking the entire chain of thousands of blocks below it in order to guarantee that the UTXO is not spent, whereas an SPV node checks how deep the block is buried by a handful of blocks above it.

To get the block headers, SPV nodes use a getheaders message instead of getblocks. The responding peer will send up to 2,000 block headers using a single headers message. The process is otherwise the same as that used by a full node to retrieve full blocks. SPV nodes also set a filter on the connection to peers, to filter the stream of future blocks and transactions sent by the peers. Any transactions of interest are retrieved using a getdata request. The peer generates a tx message containing the transactions, in response. Figure 7 shows the synchronization of block headers.

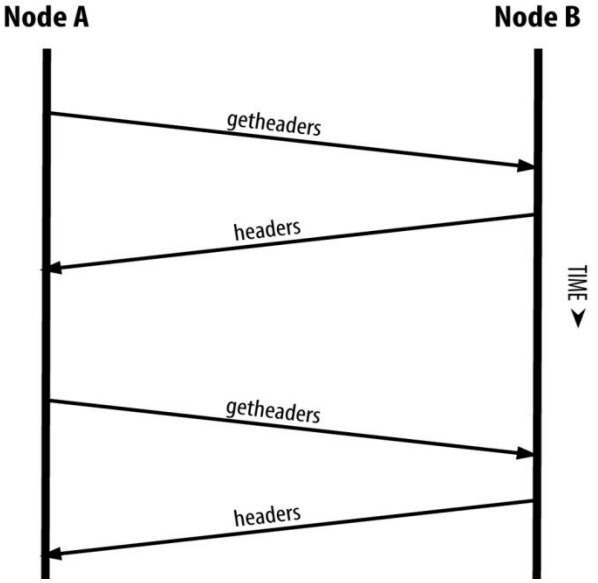

Figure 7. SPV node synchronizing the block headers

Because SPV nodes need to retrieve specific transactions in order to selectively verify them, they also create a privacy risk. Unlike full blockchain nodes, which collect all transactions within each block, the SPV node's requests for specific data can inadvertently reveal the addresses in their wallet. For example, a third party monitoring a network could keep track of all the transactions requested by a wallet on an SPV node and use those to associate bitcoin addresses with the user of that wallet, destroying the user's privacy.

Shortly after the introduction of SPV/lightweight nodes, the bitcoin developers added a feature called bloom filters to address the privacy risks of SPV nodes. Bloom filters allow SPV nodes to

receive a subset of the transactions without revealing precisely which addresses they are interested in, through a filtering mechanism that uses probabilities rather than fixed patterns.

Bloom Filters

A bloom filter is a probabilistic search filter, a way to describe a desired pattern without specifying it exactly. Bloom filters offer an efficient way to express a search pattern while protecting privacy. They are used by SPV nodes to ask their peers for transactions matching a specific pattern, without revealing exactly which addresses they are searching for.

In our previous analogy, a tourist without a map is asking for directions to a specific address, "23 Church St." If she asks strangers for directions to this street, she inadvertently reveals her destination. A bloom filter is like asking, "Are there any streets in this neighborhood whose name ends in R-C-H?" A question like that reveals slightly less about the desired destination than asking for "23 Church St." Using this technique, a tourist could specify the desired address in more detail as "ending in U-R-C-H" or less detail as "ending in H." By varying the precision of the search, the tourist reveals more or less information, at the expense of getting more or less specific results. If she asks a less specific pattern, she gets a lot more possible addresses and better privacy, but many of the results are irrelevant. If she asks for a very specific pattern, she gets fewer results but loses privacy.

Bloom filters serve this function by allowing an SPV node to specify a search pattern for transactions that can be tuned toward precision or privacy. A more specific bloom filter will produce accurate results, but at the expense of revealing what addresses are used in the user's wallet. A less specific bloom filter will produce more data about more transactions, many irrelevant to the node, but will allow the node to maintain better privacy.

An SPV node will initialize a bloom filter as "empty" and in that state the bloom filter will not match any patterns. The SPV node will then make a list of all the addresses in its wallet and create a search pattern matching the transaction output that corresponds to each address. Usually, the search pattern is a pay-to-public-key-hash script that is the expected locking script that will be present in any transaction paying to the public-key-hash (address). If the SPV node is tracking the balance of a P2SH address, the search pattern will be a pay-to-script-hash script, instead. The SPV node then adds each of the search patterns to the bloom filter, so that the bloom filter can recognize the search pattern if it is present in a transaction. Finally, the bloom filter is sent to the peer and the peer uses it to match transactions for transmission to the SPV node.

Bloom filters are implemented as a variable-size array of N binary digits (a bit field) and a variable number of M hash functions. The hash functions are designed to always produce an output that is between 1 and N, corresponding to the array of binary digits. The hash functions are generated deterministically, so that any node implementing a bloom filter will always use the same hash functions and get the same results for a specific input. By choosing different length (N) bloom filters and a different number (M) of hash functions, the bloom filter can be tuned, varying the level of accuracy and therefore privacy.

In Figure 8, we use a very small array of 16 bits and a set of three hash functions to demonstrate how bloom filters work.

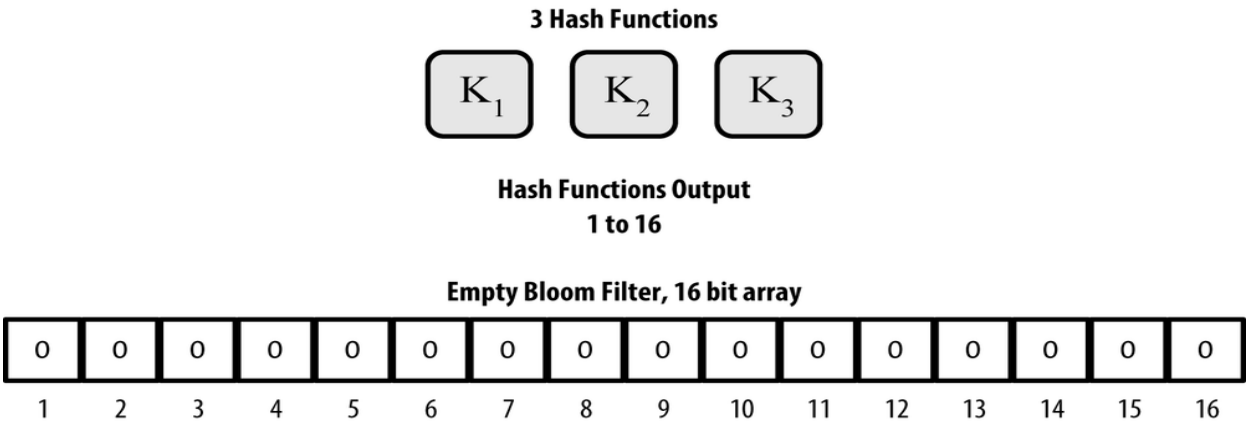

Figure 8. An example of a simplistic bloom filter, with a 16-bit field and three hash functions

The bloom filter is initialized so that the array of bits is all zeros. To add a pattern to the bloom filter, the pattern is hashed by each hash function in turn. Applying the first hash function to the input results in a number between 1 and N. The corresponding bit in the array (indexed from 1 to N) is found and set to 1, thereby recording the output of the hash function. Then, the next hash function is used to set another bit and so on. Once all M hash functions have been applied, the search pattern will be "recorded" in the bloom filter as M bits that have been changed from 0 to 1.

Figure 9 is an example of adding a pattern "A" to the simple bloom filter shown in Figure 8.

Adding a second pattern is as simple as repeating this process. The pattern is hashed by each hash function in turn and the result is recorded by setting the bits to 1. Note that as a bloom filter is filled with more patterns, a hash function result might coincide with a bit that is already set to 1, in which case the bit is not changed. In essence, as more patterns record on overlapping bits, the bloom filter starts to become saturated with more bits set to 1 and the accuracy of the filter decreases. This is why the filter is a probabilistic data structure—it gets less accurate as more

patterns are added. The accuracy depends on the number of patterns added versus the size of the bit array (N) and number of hash functions (M). A larger bit array and more hash functions can record more patterns with higher accuracy. A smaller bit array or fewer hash functions will record fewer patterns and produce less accuracy.

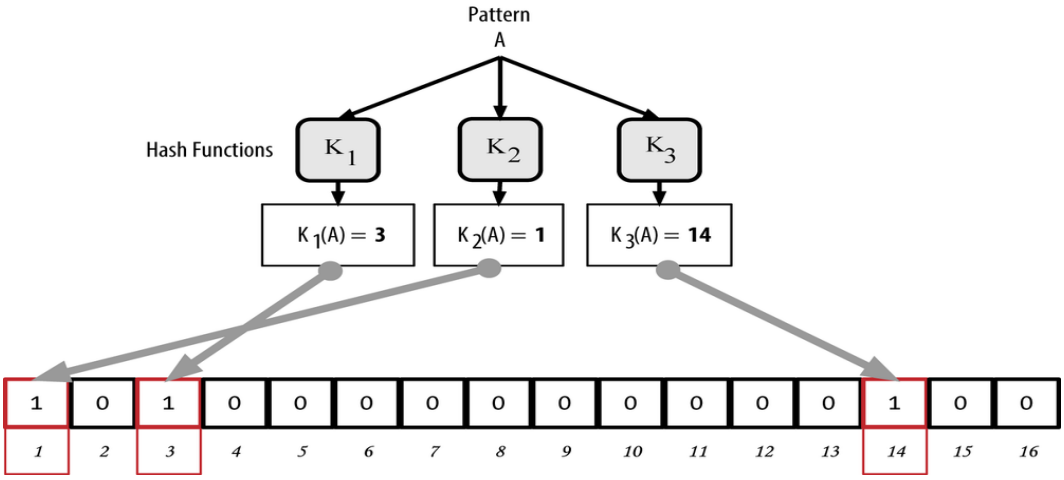

Figure 9. Adding a pattern "A" to our simple bloom filter

Figure 10 is an example of adding a second pattern "B" to the simple bloom filter.

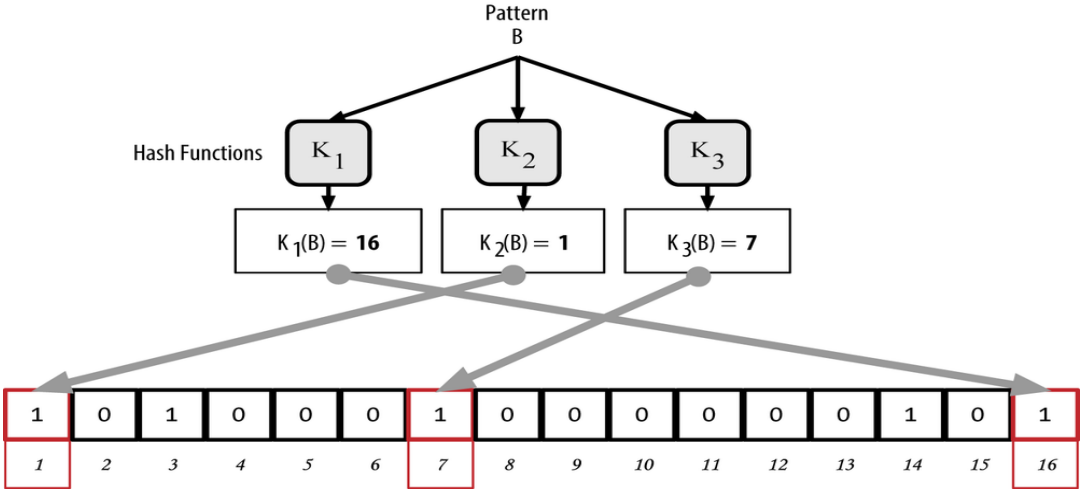

Figure 10. Adding a second pattern "B" to our simple bloom filter

To test if a pattern is part of a bloom filter, the pattern is hashed by each hash function and the resulting bit pattern is tested against the bit array. If all the bits indexed by the hash functions are set to 1, then the pattern is probably recorded in the bloom filter. Because the bits may be set because of overlap from multiple patterns, the answer is not certain, but is rather probabilistic. In simple terms, a bloom filter positive match is a "Maybe, Yes."

Figure 11 is an example of testing the existence of pattern “X” in the simple bloom filter. The corresponding bits are set to 1, so the pattern is probably a match.

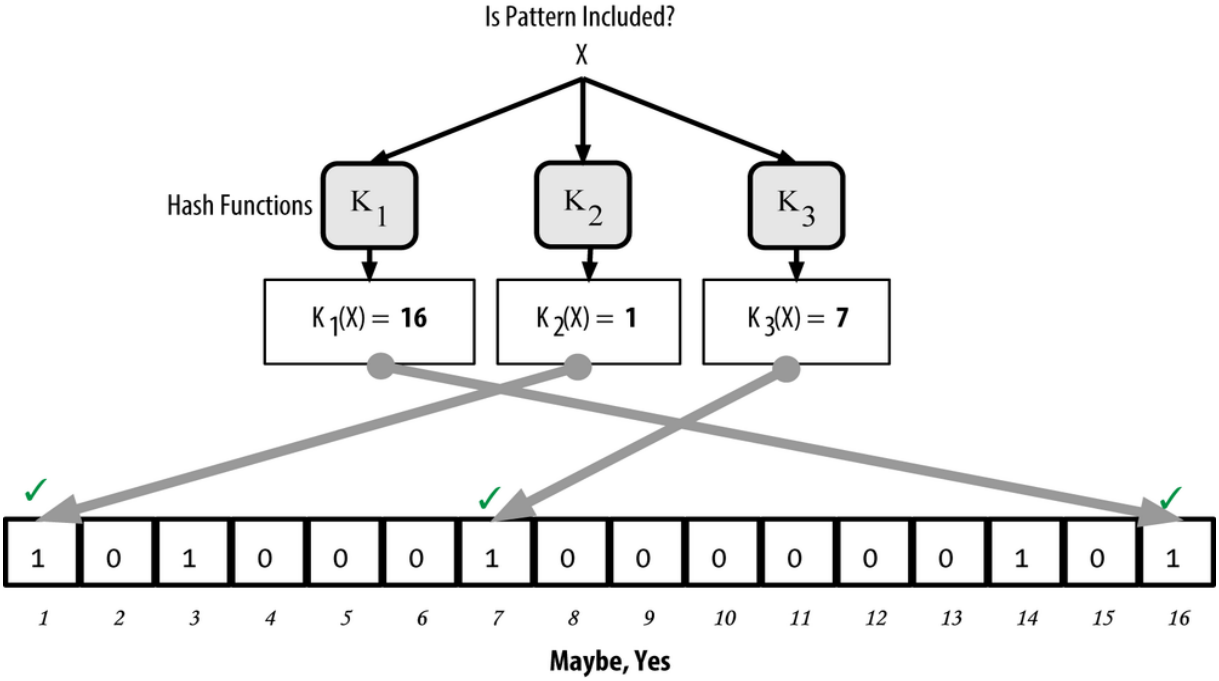

Figure 11. Testing the existence of pattern “X” in the bloom filter. The result is probabilistic positive match, meaning “Maybe.”

On the contrary, if a pattern is tested against the bloom filter and any one of the bits is set to 0, this proves that the pattern was not recorded in the bloom filter. A negative result is not a probability, it is a certainty. In simple terms, a negative match on a bloom filter is a “Definitely Not!”

Figure 12 is an example of testing the existence of pattern “Y” in the simple bloom filter. One of the corresponding bits is set to 0, so the pattern is definitely not a match.

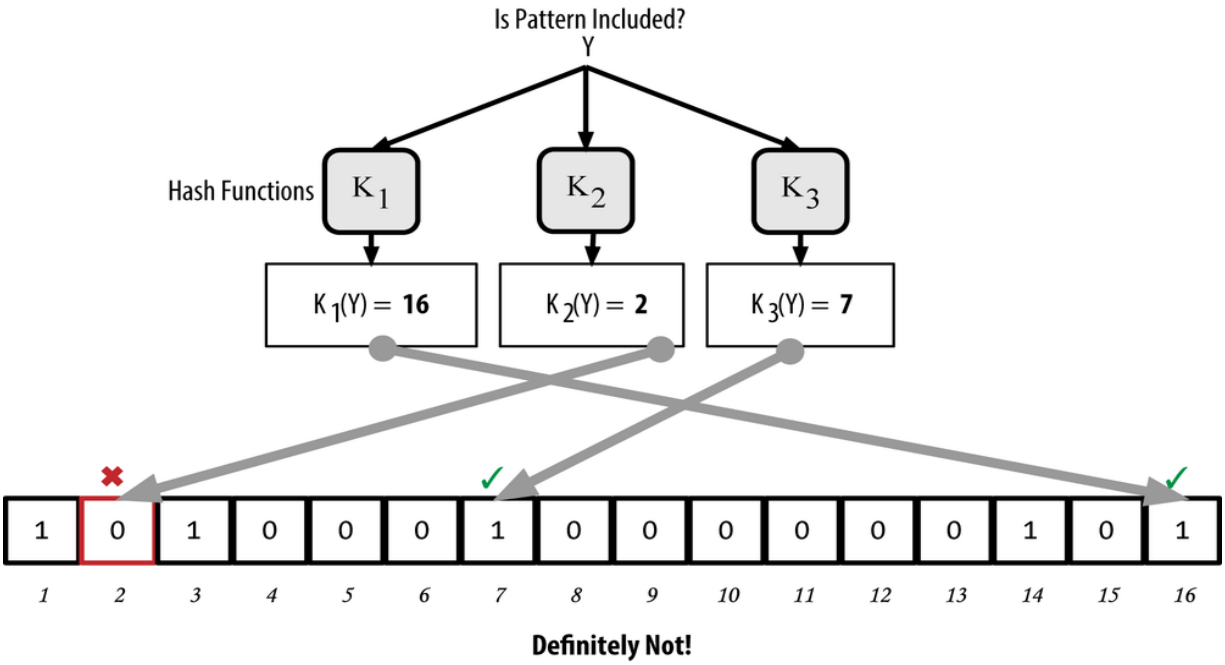

Figure 12. Testing the existence of pattern "Y" in the bloom filter. The result is a definitive negative match, meaning "Definitely Not!"

Bitcoin's implementation of bloom filters is described in Bitcoin Improvement Proposal 37 (BIP0037). See Appendix B or visit GitHub.

Bloom Filters and Inventory Updates

Bloom filters are used to filter the transactions (and blocks containing them) that an SPV node receives from its peers. SPV nodes will create a filter that matches only the addresses held in the SPV node's wallet. The SPV node will then send a `filterload` message to the peer, containing the bloom filter to use on the connection. After a filter is established, the peer will then test each transaction's outputs against the bloom filter. Only transactions that match the filter are sent to the node.

In response to a `getdata` message from the node, peers will send a `merkleblock` message that contains only block headers for blocks matching the filter and a merkle path (see Merkle Trees) for each matching transaction. The peer will then also send `tx` messages containing the transactions matched by the filter.

The node setting the bloom filter can interactively add patterns to the filter by sending a `filteradd` message. To clear the bloom filter, the node can send a `filterclear` message. Because it is not

possible to remove a pattern from a bloom filter, a node has to clear and resend a new bloom filter if a pattern is no longer desired.

Transaction Pools

Almost every node on the bitcoin network maintains a temporary list of unconfirmed transactions called the memory pool, mempool, or transaction pool. Nodes use this pool to keep track of transactions that are known to the network but are not yet included in the blockchain. For example, a node that holds a user's wallet will use the transaction pool to track incoming payments to the user's wallet that have been received on the network but are not yet confirmed.

As transactions are received and verified, they are added to the transaction pool and relayed to the neighboring nodes to propagate on the network.

Some node implementations also maintain a separate pool of orphaned transactions. If a transaction's inputs refer to a transaction that is not yet known, such as a missing parent, the orphan transaction will be stored temporarily in the orphan pool until the parent transaction arrives.

When a transaction is added to the transaction pool, the orphan pool is checked for any orphans that reference this transaction's outputs (its children). Any matching orphans are then validated. If valid, they are removed from the orphan pool and added to the transaction pool, completing the chain that started with the parent transaction. In light of the newly added transaction, which is no longer an orphan, the process is repeated recursively looking for any further descendants, until no more descendants are found. Through this process, the arrival of a parent transaction triggers a cascade reconstruction of an entire chain of interdependent transactions by re-uniting the orphans with their parents all the way down the chain.

Both the transaction pool and orphan pool (where implemented) are stored in local memory and are not saved on persistent storage; rather, they are dynamically populated from incoming network messages. When a node starts, both pools are empty and are gradually populated with new transactions received on the network.

Some implementations of the bitcoin client also maintain a UTXO database or UTXO pool, which is the set of all unspent outputs on the blockchain. Although the name "UTXO pool" sounds similar to the transaction pool, it represents a different set of data. Unlike the transaction and orphan pools, the UTXO pool is not initialized empty but instead contains millions of entries of unspent

transaction outputs, including some dating back to 2009. The UTXO pool may be housed in local memory or as an indexed database table on persistent storage.

Whereas the transaction and orphan pools represent a single node's local perspective and might vary significantly from node to node depending upon when the node was started or restarted, the UTXO pool represents the emergent consensus of the network and therefore will vary little between nodes. Furthermore, the transaction and orphan pools only contain unconfirmed transactions, while the UTXO pool only contains confirmed outputs.

Alert Messages

Alert messages are a seldom used function, but are nevertheless implemented in most nodes. Alert messages are bitcoin's "emergency broadcast system," a means by which the core bitcoin developers can send an emergency text message to all bitcoin nodes. This feature is implemented to allow the core developer team to notify all bitcoin users of a serious problem in the bitcoin network, such as a critical bug that requires user action. The alert system has only been used a handful of times, most notably in early 2013 when a critical database bug caused a multiblock fork to occur in the bitcoin blockchain.

Alert messages are propagated by the alert message. The alert message contains several fields, including:

ID
: An alert identified so that duplicate alerts can be detected

Expiration
: A time after which the alert expires

RelayUntil
: A time after which the alert should not be relayed

MinVer, MaxVer
: The range of bitcoin protocol versions that this alert applies to

subVer
: The client software version that this alert applies to

Priority
: An alert priority level, currently unused

Alerts are cryptographically signed by a public key. The corresponding private key is held by a few select members of the core development team. The digital signature ensures that fake alerts will not be propagated on the network.

Each node receiving this alert message will verify it, check for expiration, and propagate it to all its peers, thus ensuring rapid propagation across the entire network. In addition to propagating the alert, the nodes might implement a user interface function to present the alert to the user.

In the Bitcoin Core client, the alert is configured with the command-line option -alertnotify, which specifies a command to run when an alert is received. The alert message is passed as a parameter to the alertnotify command. Most commonly, the alertnotify command is set to generate an email message to the administrator of the node, containing the alert message. The alert is also displayed as a pop-up dialog in the graphical user interface (bitcoin-Qt) if it is running.

Other implementations of the bitcoin protocol might handle the alert in different ways. Many hardware-embedded bitcoin mining systems do not implement the alert message function because they have no user interface. It is strongly recommended that miners running such mining systems subscribe to alerts via a mining pool operator or by running a lightweight node just for alert purposes.